iBRAIN2: Automated analysis and data handling for RNAi screens

Rouilly, Vincent ; Pujadas, Eva ; Hullár, Béla ; Balázs, Csabas ; Kunszt, Peter ; Podvinec, Michael

Abstract: We report on the implementation of a software suite dedicated to the management and analysis of large scale RNAi High Content Screening (HCS). We describe the requirements identified amongst our different users, the supported data flow, and the implemented software. Our system is already supporting productively three different laboratories operating in distinct IT infrastructures. The system was already used to analyze hundreds of RNAi HCS plates.

DOI: https://doi.org/10.3233/978-1-61499-054-3-205

iBRAIN2: Automated analysis and data handling for RNAi screens

Vincent Rouilly [a,b,], Eva Pujadas[a,b] , Béla Hullár[b,c,e], Csaba Balázs[d, e] ,Peter Kunszt [b,e]
Michael Podvinec [a 1]
[a] *Research IT, Biozentrum, University of Basel, Switzerland.*
[b] *SyBIT, systemsx.ch, Switzerland.*
[c] *Institute of Molecular Systems Biology,*
[d] *Light Microscopy Center,*
[e] *ETH Zurich, Switzerland*

Abstract. We report on the implementation of a software suite dedicated to the management and analysis of large scale RNAi High Content Screening (HCS). We describe the requirements identified amongst our different users, the supported data flow, and the implemented software. Our system is already supporting productively three different laboratories operating in distinct IT infrastructures. The system was already used to analyze hundreds of RNAi HCS plates.

Keywords. Workflow management system, High Content Screening, RNAi

Introduction

In systems biology, RNAi-based High Content Screening (HCS) has become an indispensable tool to study cellular phenomena on a large scale [8,12]. These perturbation experiments can deliver genome-scale information correlating cellular phenotypes with specific environmental or internal challenges, and generated data sets are common starting points for the modeling of cellular interaction networks [2,3]. At the same time, HCS experiments produce large datasets that require complex analysis procedures, as well as reliable and traceable processing, and thus present significant IT challenges to researchers [5,7].

The InfectX consortium (a SystemsX.ch Research and Technology Development project) is a collaborative project of several Swiss research groups that studies genes involved in early infection events across 7 different pathogens [9]. InfectX is running multiple comparative genome-wide RNAi screening campaigns at different locations in Switzerland, producing a very large amount of image data (dozens of terrabytes). In this paper we report on the software we have developed and deployed to support the analysis and data management requirements of InfectX and the collaborating groups.

Our software system for RNAi HCS integrates automated analysis and customizable data management. It enables robust and complex parallel processing on computer cluster infrastructure and allows for reliable storage of primary and resulting data. We also discuss the software environment that was integrated for automated screening data analysis.

1. Requirements for an HCS data processing system

The InfectX consortium consists of eleven research groups at different universities and plans more than a dozen genome-wide RNAi screens to study host contributions to pathogen entry in early infection. These experiments are done in seven of the member groups, and three member laboratories were selected to handle imaging and data analysis. As each of these operate at a different university, they all have different high performance computing (HPC) resources and data storage solutions available.
Furthermore, a large diversity of user profiles had to be taken into account in the design of the system, from experimental biologists to image processing specialists, statisticians and modelers. In the following, we describe the requirements for such a software solution from the perspective of three abstracted research environments:

1. Analysis Research Lab: Developing novel image analysis and statistical methods to analyze screening data.
2. Biology Research Lab: Primarily interested in analysis results to drive further, detailed biological experiments. Their main interest is in developing and applying robust, standardized, and reproducible analysis workflows.
3. Imaging/HCS Technology Platform: As a service provider, such a laboratory must accommodate large throughput and lots of different users.

An analysis research lab does active research in RNAi screening methods, focusing on novel phenotype characterization and new statistical data analysis methods. Image- and data analysis routines in the current processing workflows are constantly changing, and therefore, high flexibility is a key requirement. They are less interested in the infrastructure details, and are highly proficient in the analysis "mechanism". They require data at their fingertips for in-depth analysis of certain features and the ability to re-launch large-scale analysis of existing data with the latest method easily. Most of the development for image analysis is done in scripting languages like Matlab and Python, but they also need to integrate custom programs written in any language. The workflows itself are modified very frequently as they are subject to active research.

A biology research lab requires the ability to use and re-use robust and standard processing workflows usable by the whole consortium. Using cutting-edge methods developed by the previous laboratory type, this laboratory values reproducibility and comparability above all, to allow different screening experiments within the consortium to be compared. The applied workflows are complex with many steps, and as they continue to evolve, it is critical to keep track of different versions, and to always be able to trace data provenance. It is also important to be able to re-analyze large amounts of data with a new workflow for comparative analysis.

The final type of requirements describes the needs of a microscopy technology platform that operates as a service. It provides RNAi screening services to a large number of research groups. Depending on the project, they need to be able to select the right kind of workflow. They have an already established in-house infrastructure (specific storage and HPC resources), with which analysis tools need to integrate. Their main interest is to relieve staff by automating various types of "standard" workflows while providing a faster turnaround of their services and ensuring reproducibility of the services offered. Here, robustness and transparent error handling are very important to be able to optimize the time to react to issues.

Taken together, these requirements have shaped the design of our solution. In the next section we describe the specificities of an RNAi HCS data flow and elaborate how the needs for flexibility, robustness, and standardization were addressed.

2. RNAi HCS Data and Data Flow Requirements

A single genome-wide RNAi screen typically consists of seventy 384-well plates, or ~27,000 individual perturbation experiments. High-throughput microscopes acquire raw fluorescent images (multiple channels) from each well at multiple locations. This large set of primary data needs to be immediately safely stored in the 'project store', where it is available for the duration of the project and needs to be made available on a HPC resource to perform a complex series of intensive processing steps. From each step, results may need to be safely stored in the *project store*, annotated regarding their provenance. Finally, when all analysis is done, data need to be published and/or archived.

The list below summarizes the key steps in processing RNAi HCS data:

1. **Acquisition**: Raw images are produced by automated microscopes at a rate of 5Gb/h/instrument. Imaging one 384-well plate results, on average, in 20Gb of raw data contained in more than 12,000 TIFF images. A full genome-wide screen comprises 70 such plates.
2. **Data storage and preparation**: Data is moved to a long-term storage system where it is stored with sufficient redundancy and availability for the duration of the project (1-5years). We call this the *project store*.
3. **Data staging to HPC**. In some infrastructures, the *project store* is directly accessible from the cluster. More often, a temporary copy of the input data is fetched from the *project store* and made accessible on the *scratch area* of the HPC cluster file system.
4. **Data analysis.** This is done on a HPC cluster or large server. Over the last decade, a rich ecosystem of RNAi HCS tools has emerged. We distinguish five commonly used operations during the analysis and list the most common software we had to integrate in our workflows. All steps except for the last statistical analysis are embarrassingly parallel.
    4.1. **Image quality assessment and pre-processing**. This step involves intensive image processing computation, implemented in Matlab or C++.
    4.2. **Image segmentation** to detect cellular components (nuclei, cytoplasm, pathogens, etc.) This step is typically performed with specialized image processing software toolkits, such as CellProfiler [4, 17] or ImageJ [13].
    4.3. **Cell feature extraction**. From segmented images, quantitative metrics are evaluated in order to parametrize each image's objects. This step is typically performed with specialized image processing software toolkits, such as CellProfiler or ImageJ.
    4.4. **Phenotype characterization**. Using a machine learning approach, each detected cell is labeled with a biologically meaningful phenotype. User

input is required in the training [10,18]. We have integrated the Matlab-based 'CellClassifier' software, developed within the consortium [15].

4.5. **Statistical analysis and hit-list generation**. Following data normalization and an experiment-wide comparison, a list of genes with the highest impact can be extracted [14,19].

5. **Result data storage**: From each step of the workflow, a subset of the computed results (*final results*) can be moved from the cluster file system to the *project store*, intermediate results are removed after workflow execution.

6. **Publication and archiving**: At the end of a project or for a publication, a subset of the data is made available publicly, and data worth keeping is transferred to a long-term archival facility.

This data analysis workflow highlights the importance of a data flow that transparently integrates a safe, long-term *project store* with an efficient short-term HPC file system. It also defines a system able to schedule and monitor jobs executing custom code and established RNAi third-party tools (CellProfiler, Matlab, R, etc.) on an HPC resource. A further critical requirement, due to the scale and distributed nature of the project, is the automatic annotation of workflow results regarding the specifics of the workflow that produced them (software versions and analysis parameters)

3. iBRAIN2: An automated workflow system for RNAi experiments

One of the consortium partners had previously developed an analysis tool for RNAi screens, called iBRAIN, to deal with the group's needs and research projects [16]. In collaboration with the original authors, we have taken up the task of re-engineering this tool so that it could become a versatile platform to be shared across the different InfectX partners. The goal of this approach was to use standard software engineering

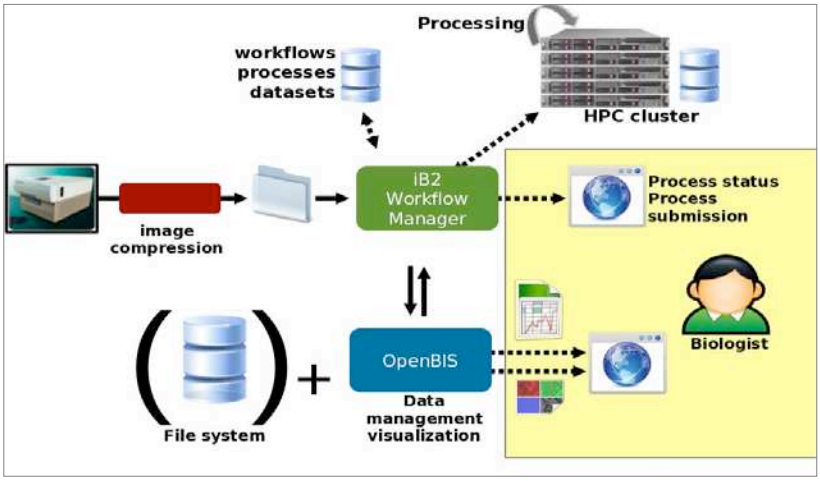

***Figure 1**: The iBRAIN2 workflow manager can flexibly integrate into a local infrastructure with a HPC resource, as well as OpenBIS as a data management system. Users can manage their workflows through a web interface.*

methods, tools and open-source libraries and software to develop a robust and modular workflow system that handles the full life-cycle of RNAi data. This is why our tool is called iBRAIN2.

As mentioned in the previous sections, a need to adapt to different IT infrastructure with specific HPC and storage resources was paramount. iBRAIN2 was designed to give easy access to HPC and data management resources by abstracting the required operations and providing corresponding interfaces. Additionally, iBRAIN2 aimed at providing a workflow management solution so that researchers could put together established RNAi tools – or their own novel tools - in a flexible and reproducible manner. As validation is a key element in standardization, each workflow module is allowed to specify not only input data, but also assertions about what constitutes valid output.

iBRAIN2 is implemented in Java. It consists of a scheduler daemon and a web interface. The daemon is designed as a state machine with persistence built in through a SQL database. The daemon can process multiple datasets in parallel, taking care of all the necessary HPC and storage transactions. A web interface allows users to register their assays (consisting of a set of plates) and workflows. It is also through the web interface that users can trigger and monitor analysis workflows. To provide flexibility, a number of interfaces are built-in in the iBRAIN2 design. We list below the details of theses interfaces and demonstrate how they allow our software to be not only flexible in response to different infrastructure contexts, but also modular in integrating all the necessary analytical tools to perform a full RNAi processing [FIGURE 1].

3.1 User interaction: Web interfaces and programmatic access

A web interface allows users (from bioinformaticians to statisticians) to interact with the system. Here, assays (logical groups of experimental datasets) and workflows can be registered. Assays can be configured to execute default workflows as soon as a dataset belonging to it is discovered by the system. Computational processes can be triggered and monitored, as execution proceeds. Information is available from a general overview down to details about each workflow module [FIGURE 2].

Recently, we have extended iBRAIN2 with a REST-based interface that allows programmatic access to its functionality.

3.2 Flexible integration with local HPC and storage infrastructure

In this project, enabling pre-existing local IT infrastructure was essential, as mentioned above. This goal was achieved through dedicated Java interfaces for HPC local resource management systems (LRMS) and storage component integration.

The iBRAIN2 HPC interface defines the core methods for job submission and monitoring as well as the different job statuses that the system has to distinguish. It directly supports all OGF DRMAA V1-compliant [6] LRMS, as well as Platform LSF [11]. During the iBRAIN2 installation, administrators can add additional parameterization, e.g. defining a mapping between the queue type definitions in iBRAIN2 and existing cluster queues (thus targeting jobs to the appropriate resources).

iBRAIN2 takes care of automatic storage and retrieval of both raw and result data. For this purpose, we have designed a StorageProvider interface that allows iBRAIN2 to be integrated with different data management solutions. Currently, two

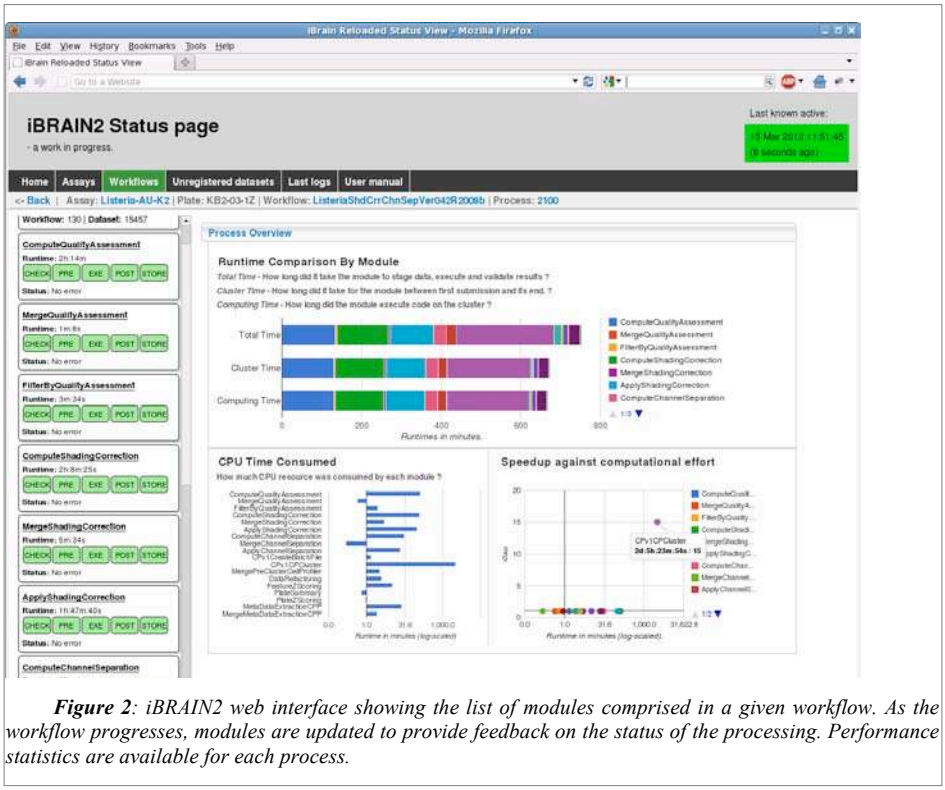

***Figure 2**: iBRAIN2 web interface showing the list of modules comprised in a given workflow. As the workflow progresses, modules are updated to provide feedback on the status of the processing. Performance statistics are available for each process.*

functional implementations of this interface exist: *File system* and *OpenBIS*. *File system* storage uses the file system provided by the operating system to store raw and result data in a hierarchical, logical directory tree. Local and network drives are supported by this StorageProvider.

The *OpenBIS* StorageProvider provides an interface to the Open Biology Information System (OpenBIS), another SyBIT project that focuses on data management, annotation with meta data, retrieval and visualization, and which is optimized for large-scale storage [1]. OpenBIS comes with a set of auxiliary tools and services: In addition to the metadata store, a data mover service, and a data storage server manage access to several different data stores with different access policies, providing an integrated view to the users. For large production environments, we strongly recommend using OpenBIS rather than the file system for data storage.

3.3 Flexible execution of processing workflows

In previous sections, we have described both the need for flexible, modular workflows, as well as the existence of a large pool of software commonly used by scientists to analyze RNAi data. From image processing to machine learning and statistical analysis, iBRAIN2 provides the means to orchestrate complex processing workflows in a

traceable and reproducible way. The system was built with resilience in mind. It keeps track of the workflow steps in its local state database and can take predefined actions on various types of failures in the workflow. For example it will re-submit a cluster job in case it failed.

iBRAIN2 workflows are defined in an XML document, describing each step of the processing. A flexible mapping of the command line of any software allows tracking all the parameters and software versions used for a given processing. The description also includes the way to define input-output dependencies between each step of the workflow, as well as a description of the computed data to be safely stored in the *project store*. Moreover, assertions about expected output data can be specified and are used to validate successful execution. These XML workflows can be exchanged between the different centers to insure a consistent and standard processing of the data in the project.

Currently, three iBRAIN2 production installations exist within the InfectX project, each connected to its own openBIS instance. Two of these have fully entered large-scale production mode and use quite different local storage setup and cluster infrastructures (SGE and LSF queuing systems, respectively). To date, the system has processed more than 50TB of raw data. More than 30 different workflows are registered, and dozens of key users rely on the system every day.

4. Conclusions and Future Work

RNAi HCS experiments provide a revolutionary way to explore at the genome level the complexity of processes of living matter. However, they also generate a wealth of data that requires intensive post-processing, as well as rigorous curation. In this paper, we have described the requirements arising from a multi-center project to analyze, contrast and compare datasets from genome-wide screens. We further describe the iBRAIN2 system that was developed to address a number of these requirements. Key factors in its development was the ability to flexibly react to changing requirements, as is typical in scientific projects. In fact, the system has been in continuous production use at least at one site throughout all of its development. This additional requirement of being in production from a very early stage has influenced the design of the system to no small extent. This has led to a very modular design, which allows the software to be adapted to any IT infrastructure with regards to HPC and storage resources.

iBRAIN2 is still under active development within SyBIT, we continue to refine its interactions with OpenBIS and update its functionalities depending on new requirements emerging from the scientific requirements. The scientific case driving iBRAIN2 comes from the InfectX research project which has not yet concluded and will publish its results later this year. Then we will be also able to report on detailed usage and benchmarking of iBRAIN2.

The successful management and analysis of high-content data clearly relies not only on a single tool, however, but extends throughout the full data flow. Therefore, we are also taking an active role in trying to smoothen and standardize the data flow, which to date requires many sub-optimal and ad hoc interventions. Currently, the XML workflow descriptions are specifically geared to our community's needs and one future goal is the

conversion into standardized workflow description languages and to compare and adopt other existing technologies. Moreover, we work with instrument vendors in the high-content field to better standardize image formats for initial data production, like OME-TIFF, and to provide data in container formats (e.g. HDF5) for better manageability.

Software availability

The iBRAIN2 software and its source code are available for free. The latest iBRAIN2 release and documentation can be found at http://sybit.net/projects/current_projects/iBRAIN2/

Acknowledgements

We wish to thank Berend Snijder, University of Zürich, Switzerland as the developer of the original iBRAIN. We thank Mario Emmenlauer and Pauli Rämö, University of Basel, Switzerland, for regular discussions and feedback around the development of data analysis workflows. We are also very thankful to all our collaborators from the InfectX project, for their continuous and valuable feedback during the development of our software. We also would like to thank the OpenBIS team at the Center of Information Science and Databases of the ETH Zürich in Basel, led by Bernd Rinn. Finally, we acknowledge the help of Rainer Pöhlmann and the BC2 bioinformatics system administrators regarding HPC infrastructure questions. This work was supported by SystemsX.ch in the context of the SyBIT and InfectX.ch projects.

References

[1] Bauch, A.; Adamczyk, I.; Buczek, P.; Elmer, F.-J.; Enimanev, K.; Glyzewski, P.; Kohler, M.; Pylak, T.; Quandt, A.; Ramakrishnan, C.; Beisel, C.; Malmstrom, L.; Aebersold, R. & Rinn, B. (2011), 'openBIS: a flexible framework for managing and analyzing complex data in biology research.', *BMC Bioinformatics* 12(1), 468.

[2] Boutros, M. & Ahringer, J. (2008), 'The art and design of genetic screens: RNA interference.', *Nat Rev Genet* **9**(7), 554--566.

[3] Boutros, M.; Brás, L. P. & Huber, W. (2006), 'Analysis of cell-based RNAi screens.', *Genome Biol* **7**(7), R66.

[4] Carpenter, A. E.; Jones, T. R.; Lamprecht, M. R.; Clarke, C.; Kang, I. H.; Friman, O.; Guertin, D. A.; Chang, J. H.; Lindquist, R. A.; Moffat, J.; Golland, P. & Sabatini, D. M. (2006), 'CellProfiler: image analysis software for identifying and quantifying cell phenotypes.', *Genome Biol* **7**(10), R100.

[5] Conrad, C. & Gerlich, D. W. (2010), 'Automated microscopy for high-content RNAi screening.', *J Cell Biol* **188**(4), 453--461.

[6] DRMAA, http://www.drmaa.org/

[7] Ghosh, S.; Matsuoka, Y.; Asai, Y.; Hsin, K.-Y. & Kitano, H. (2011), 'Software for systems biology: from tools to integrated platforms.', *Nat Rev Genet* **12**(12), 821--832.
[8] Haney, S. A.; LaPan, P.; Pan, J. & Zhang, J. (2006), 'High-content screening moves to the front of the line.', *Drug Discov Today* **11**(19-20), 889--894.
[9] InfectX consortium, www.infectx.org.', InfectX, systemsx.ch, Switzerland.
[10] Jones, T. R.; Kang, I. H.; Wheeler, D. B.; Lindquist, R. A.; Papallo, A.; Sabatini, D. M.; Golland, P. & Carpenter, A. E. (2008), 'CellProfiler Analyst: data exploration and analysis software for complex image-based screens.', *BMC Bioinformatics* **9**, 482.
[11] Platform LSF. http://www.platform.com/workload-management/high-performance-computing.
[12] Montgomery, M. K. (2006), 'RNA interference: unraveling a mystery.', *Nat Struct Mol Biol* **13**(12), 1039--1041.
[13] Rasband, W. (1997-2011), 'ImageJ, U. S. National Institutes of Health, http://imagej.nih.gov/ij/.', U. S. National Institutes of Health, Bethesda, Maryland, USA.
[14] Rieber, N.; Knapp, B.; Eils, R. & Kaderali, L. (2009), 'RNAither, an automated pipeline for the statistical analysis of high-throughput RNAi screens.', *Bioinformatics* **25**(5), 678--679.
[15] Rämö, P.; Sacher, R.; Snijder, B.; Begemann, B. & Pelkmans, L. (2009), 'CellClassifier: supervised learning of cellular phenotypes.', *Bioinformatics* **25**(22), 3028--3030.
[16] Snijder, B.; Sacher, R.; Ramo, P.; Damm, E.-M.; Liberali, P. & Pelkmans, L. (2009), 'Population context determines cell-to-cell variability in endocytosis and virus infection', *Nature* **461**(7263), 520--523.
[17] Vokes, M. S. & Carpenter, A. E. (2008), 'Using CellProfiler for automatic identification and measurement of biological objects in images.', *Curr Protoc Mol Biol* **Chapter 14**, Unit 14.17.
[18] Wang, J.; Zhou, X.; Bradley, P. L.; Chang, S.-F.; Perrimon, N. & Wong, S. T. C. (2008), 'Cellular phenotype recognition for high-content RNA interference genome-wide screening.', *J Biomol Screen* **13**(1), 29--39.
[19] Wang, X.; Terfve, C.; Rose, J. C. & Markowetz, F. (2011), 'HTSanalyzeR: an R/Bioconductor package for integrated network analysis of high-throughput screens.', *Bioinformatics* **27**(6), 879--880.